POMPTON LAKES SCHOOL DISTRICT

Introduction to Computer Science

Semester Course

COURSE OF STUDY

June 2018

Submitted by
The Mathematics Department

Dr. Paul Amoroso, *Superintendent*
Mr. Vincent Przybylinski, *Principal*
Mr. Anthony Mattera, *Vice Principal*
Mr. Thomas Salus, *Board of Education President*
Mrs. Stephanie Shaw, *Board of Education Vice President*

Board Members
Mrs. Traci Cioppa, Mr. Robert Cruz, Mrs. Eileen Horn,
Ms. Kelly Norris, Mr. Karl Roman, Mr. Scott Santers,
Mrs. Nancy Schwartz and Mr. Tim Troast

Unit Overview

Content Area: Technology

Unit Title: Unit 1: Programming With Basic Karel

Target Course/Grade Level: Introduction to Computer Science/11 and 12

Unit Summary

Programming with Karel utilizes a dog named Karel to introduce the student to basic computer programming. The student will develop code to have Karel perform various tasks contained in a grid of streets and avenues. Due to Karel's limited movement options, the student will learn to write basic code to aid him or her to accomplish the task utilizing functions, loops, and If'/Then statements. The students will also learn programming vocabulary, syntax and debugging techniques.

Unit Rationale

The first unit, Programming with Karel, introduces the student to the basics of programming. Before one can code effectively, one must understand flowcharts and top down design. Developing a programming style and mindset will allow the student to develop code to perform simple tasks like having a dog move through a neighborhood. By the end of the unit the student will be exposed to functions, If/Then statements and loops. Debugging and computer syntax will also be emphasized.

Student Learning Objectives

Students will be able to:

1. Structure computer programs with Top Down Design.
2. Create basic code to perform a variety of tasks.
3. Utilize functions in coding.
4. Utilize loops in coding.
5. Utilize If/Then statements in coding.
6. Create comments in code to insure that other programmers understand the program.
7. Debug a computer program.

Mathematical Practices

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

All of the content presented in this course has connections to the standards for mathematical practices.

Code # 9.3	**Career & Technical Education Content Area: 21st Century Life and Careers**
IT - PRG.1	Analyze software needs and requirements.
IT – PRG.2	Demonstrate the use of industry standard strategies and project planning.
IT – PRG.3	Analyze system and software requirements to ensure maximum operating efficiency
IT – PRG.4	Demonstrate the effective use of software development tools.
IT – PRG.5	Apply an appropriate software development process to design a software application.
IT – PRG.6	Program a computer application using the appropriate programming language.
IT – PRG.7	Demonstrate software testing procedures to ensure quality.
IT – PRG.8	Perform quality assurance tasks as part of the software development cycle.
IT – PRG.9	Perform software maintenance.
IT–PRG.10	Design, create and maintain a database.

IT – WD.4	Demonstrate the effective use of tools for digital communication production, development and project management.
IT – WD.5	Develop, administer and maintain Web applications.
IT – WD.6	Design, create and publish a digital communication product.
IT – WD.7	Evaluate the functionality of a digital communication product using industry accepted techniques and metrics.
IT – WD.8	Implement quality assurance processes to deliver quality digital communication products and services.
IT – WD.10	Comply with intellectual property laws, copyright laws and ethical practices when Creating Web/digital communications.

Unit Essential Questions	**Unit Enduring Understandings**
• How does Top Down Design help the programmer create code to perform a task? • Why are functions a vital time saving feature of computer programs? • Why can indenting and commenting create code that is easier to communicate with others?	• Top Down Design breaks down a task into simpler and simpler pieces until a level has been reached that corresponds to the primitives of the programming language to be used. • Commands and functions are the building blocks of computer programs. • Control structures can be combined to create to produce responsive and powerful programs.

Career Readiness Practices
CRP2. Apply appropriate academic and technical skills.
CRP4. Communicate clearly and effectively and with reason.
CRP6. Demonstrate creativity and innovation.
CRP7. Employ valid and reliable research strategies.
CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11. Use technology to enhance productivity.

Evidence of Learning

Summative Assessment: Each unit will involve the creation of numerous computer programs. A culminating project-based program will be included in each unit.

Formative Assessments

- Tests and quizzes
- Discussions
- Individual practice
- Explanation of examples
- Daily homework assignments
- Computer Programs

Lesson Plans

Lesson	Timeframe
Lesson 1 Introduction to Programming with Karel	2 days
Lesson 2 Karel Can't Turn Right	2 days
Lesson 3 Functions in Karel	2 days
Lesson 4 The Start Function	2 days
Lesson 5 Top Down Design and Decomposition	4 days
Lesson 6 Commenting Your Code	2 days

Lesson 7 For Loops	4 days
Lesson 8 If Statements	2 days
Lesson 9 If/Else Statements	3 days
Lesson 10 While Loops	3 days
Lesson 11 Control Structures	3 days
Lesson 12 Testing Your Program	2 days
Lesson 13 Indentation	2 days

Teacher Notes:

1. 6 days allocated for formative/summative assessments
2. Students can seek input from their peers and teachers throughout collaborative assignments and activities

Curriculum Development Resources

CodeHS :Introduction to Computer Science Course

Unit Overview

Content Area: Technology

Unit Title: Unit 2: Basic Javascript and Graphics

Target Course/Grade Level: Introduction to Computer Science/11 and 12

Unit Summary

Java has become the most important computer code in the 21st century. Many professions require a cursory knowledge of this coding language. This unit will build upon the skills learned in Unit 1. Functions, loops, and If/Then statements will be utilized with the Boolean expressions and logical operators. Basic mathematical operations will expand on the variety of problems to be solved. The addition of graphics will provide the introduction to animation and games in next unit.

Unit Rationale

Java is one of the most popular programming languages used to create Web applications and platforms. It was designed for flexibility, allowing developers to write code that would run on any machine, regardless of architecture or platform. Java is used to build applications and platforms for a number of devices, including computers, laptops, gaming consoles, Blu-ray players, car navigation systems, medical monitoring devices, parking meters, lottery terminals and smartphones. It is also a key language for networking, particularly for data centers that store and transfer Web-based data.

Student Learning Objectives

Students will be able to:

1. Utilize functions with variables.
2. Input data into a computer program to compute a number or make a decision.
3. Design a program to perform fundamental math operations.
4. Create simple graphics.
5. Create Boolean expressions and logical operators.
6. Design computer code with comparison operators.
7. Program code utilizing If/Then statements and loops.

Mathematical Practices

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

All of the content presented in this course has connections to the standards for mathematical practices.

Code # 9.3	**Career & Technical Education Content Area: 21st Century Life and Careers**
IT - PRG.1	Analyze software needs and requirements.
IT – PRG.2	Demonstrate the use of industry standard strategies and project planning.
IT – PRG.3	Analyze system and software requirements to ensure maximum operating efficiency
IT – PRG.4	Demonstrate the effective use of software development tools.
IT – PRG.5	Apply an appropriate software development process to design a software application.
IT – PRG.6	Program a computer application using the appropriate programming language.
IT – PRG.7	Demonstrate software testing procedures to ensure quality.
IT – PRG.8	Perform quality assurance tasks as part of the software development cycle.
IT – PRG.9	Perform software maintenance.
IT–PRG.10	Design, create and maintain a database.

IT – WD.4	Demonstrate the effective use of tools for digital communication production, development and project management.
IT – WD.5	Develop, administer and maintain Web applications.
IT – WD.6	Design, create and publish a digital communication product.
IT – WD.7	Evaluate the functionality of a digital communication product using industry accepted techniques and metrics.
IT – WD.8	Implement quality assurance processes to deliver quality digital communication products and services.
IT – WD.10	Comply with intellectual property laws, copyright laws and ethical practices when Creating Web/digital communications.

Unit Essential Questions	**Unit Enduring Understandings**
• How does knowledge of the Cartesian coordinate system aid the computer programmer when designing graphics? • How does the use of functions and parameters allow the computer programmer to minimize the length of a program? • Why are logical and comparison operators vital to solve computer performance tasks**?**	• Boolean values refer to either true or false, 0 or 1, and yes or no. • Logical operators allow the programmer to combine and connect Booleans in useful ways. • Comparison operators allow the computer programmer two values against one another.

Career Readiness Practices
CRP2. Apply appropriate academic and technical skills.
CRP4. Communicate clearly and effectively and with reason.
CRP6. Demonstrate creativity and innovation.
CRP7. Employ valid and reliable research strategies.
CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11. Use technology to enhance productivity.

Evidence of Learning

Summative Assessment: Each unit will involve the creation of numerous computer programs. A culminating project-based program will be included in each unit.

Formative Assessments

- Tests and quizzes
- Discussions
- Individual practice
- Explanation of examples
- Daily homework assignments
- Computer Programs

Lesson Plans

Lesson	Timeframe
Lesson 1 Hello World	2 days
Lesson 2 Variables	3 days
Lesson 3 User Input	2 days
Lesson 4 Basic Math in Javascript	3 days
Lesson 5 Using Graphics in Javascript	4 days
Lesson 6 Booleans	4 day

Lesson 7 Logical Operators	2 days
Lesson 8 Comparison Operators	4 days
Lesson 9 If Statements	4 days
Lesson 10 If/Else Statements	3 days
Lesson 11 For Loops in Javascript	4 days
Lesson 12 Random Numbers	4 days
Lesson 13 While Loops	4 days
Lesson 14 Functions and Parameters	5 days
Lesson 15 Functions and Return Values	5 days

Teacher Notes:

- 6 days allocated for formative/summative assessments
- Students can seek input from their peers and teachers throughout collaborative assignments and activities

Curriculum Development Resources

CodeHS :Introduction to Computer Science Course

Unit Overview

Content Area: Technology

Unit Title: Unit 3: Animation and Games

Target Course/Grade Level: Introduction to Computer Science/11 and 12

Unit Summary

Animation drives many computer programs. This unit will introduce the student to the basics of animation with the creation of computer games. Using the knowledge gained in the first two units, primarily code writing and graphics, the student will expand their knowledge with the introduction of Events. Events are actions that happen that the program needs to or wants to respond to. The student will design programs with both mouse and keyboard events. A culminating activity will involve the design of a computer game.

Unit Rationale

Animation is a key aspect of many types of computer programs. This is especially true of games. A good understanding how animation works will be beneficial as you design and build more complex programs. Creating flipbook animations with paper is essentially the same process as creating computer animation. This unit will culminate with the design, code writing, debugging and playing of a game similar to Breakout.

Student Learning Objectives

Students will be able to:

1. Design computer programs using mouse click events.
2. Design computer programs with keyboard events.
3. Design, debug and test a computer game.

Mathematical Practices

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

All of the content presented in this course has connections to the standards for mathematical practices.

Code # 9.3	**Career & Technical Education Content Area: 21st Century Life and Careers**
IT - PRG.1	Analyze software needs and requirements.
IT – PRG.2	Demonstrate the use of industry standard strategies and project planning.
IT – PRG.3	Analyze system and software requirements to ensure maximum operating efficiency
IT – PRG.4	Demonstrate the effective use of software development tools.
IT – PRG.5	Apply an appropriate software development process to design a software application.
IT – PRG.6	Program a computer application using the appropriate programming language.
IT – PRG.7	Demonstrate software testing procedures to ensure quality.
IT – PRG.8	Perform quality assurance tasks as part of the software development cycle.
IT – PRG.9	Perform software maintenance.
IT–PRG.10	Design, create and maintain a database.
IT – WD.4	Demonstrate the effective use of tools for digital communication production, development and project management.
IT – WD.5	Develop, administer and maintain Web applications.
IT – WD.6	Design, create and publish a digital communication product.
IT – WD.7	Evaluate the functionality of a digital communication product using industry accepted techniques and metrics.

IT – WD.8	Implement quality assurance processes to deliver quality digital communication products and services.
IT – WD.10	Comply with intellectual property laws, copyright laws and ethical practices when Creating Web/digital communications.

Unit Essential Questions	**Unit Enduring Understandings**
• How does animation improve the users interface with a computer program? • How does a computer program respond to an event whether a mouse event or a keyboard event?	• Computer animation is a 21st century adaptation of flipbook animation. • Computer user inputs are treated as an event in computer programming where the program must/needs to respond.

Career Readiness Practices

CRP2. Apply appropriate academic and technical skills.
CRP4. Communicate clearly and effectively and with reason.
CRP6. Demonstrate creativity and innovation.
CRP7. Employ valid and reliable research strategies.
CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11. Use technology to enhance productivity.

Evidence of Learning

Summative Assessment: Each unit will involve the creation of numerous computer programs. A culminating project-based program will be included in each unit.

Formative Assessments

- Tests and quizzes
- Discussions
- Individual practice
- Explanation of examples
- Daily homework assignments
- Computer Programs

Lesson Plans

Lesson	Timeframe
Lesson 1 History of Animation	2 days
Lesson 2 Mouse Click Events	8 days
Lesson 3 Drawing Lines	5 days
Lesson 4 Keyboard Events	8 days
Lesson 5 Project Breakout	10 days

Teacher Notes:

- 6 days allocated for formative/summative assessments
- Students can seek input from their peers and teachers throughout collaborative assignments and activities

Curriculum Development Resources

CodeHS :Introduction to Computer Science Course

Unit Overview

Content Area: Technology

Unit Title: Unit 4: Basic Data Structures

Target Course/Grade Level: Introduction to Computer Science/11 and 12

Unit Summary

The ability to store, organize and retrieve data is imperative in the 21st century. The unit on basic data structures will introduce the student into this very important aspect of computers. The introduction to arrays, lists, objects and grids will allow the student to create many computer programs to perform functions that are vital to the computer industry.

Unit Rationale

Statistics is the branch of mathematics that deals with the collection, organization, analysis, and interpretation of numerical data. Storing, accessing and manipulating lists of data is an essential and fundamental facet of computer programming in regards to statistics. The retrieval of non-numerical and numerical data from stored lists is also used in every business and government agency. This unit will introduce the student to the basic data structures and their manipulations.

Student Learning Objectives

Students will be able to:

1. Create an array.
2. Input, remove and change an array and its elements.
3. Find elements in an array.
4. Iterate an array.
5. Define and create an object/map and use them to associate one another.
6. Create and use the elements in sets.
7. Create and use a grid.
8. Loop over a grid.

Mathematical Practices

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

All of the content presented in this course has connections to the standards for mathematical practices.

Code # 9.3	**Career & Technical Education Content Area: 21st Century Life and Careers**
IT - PRG.1	Analyze software needs and requirements.
IT – PRG.2	Demonstrate the use of industry standard strategies and project planning.
IT – PRG.3	Analyze system and software requirements to ensure maximum operating efficiency
IT – PRG.4	Demonstrate the effective use of software development tools.
IT – PRG.5	Apply an appropriate software development process to design a software application.
IT – PRG.6	Program a computer application using the appropriate programming language.
IT – PRG.7	Demonstrate software testing procedures to ensure quality.

IT – PRG.8	Perform quality assurance tasks as part of the software development cycle.
IT – PRG.9	Perform software maintenance.
IT–PRG.10	Design, create and maintain a database.
IT – WD.4	Demonstrate the effective use of tools for digital communication production, development and project management.
IT – WD.5	Develop, administer and maintain Web applications.
IT – WD.6	Design, create and publish a digital communication product.
IT – WD.7	Evaluate the functionality of a digital communication product using industry accepted techniques and metrics.
IT – WD.8	Implement quality assurance processes to deliver quality digital communication products and services.
IT – WD.10	Comply with intellectual property laws, copyright laws and ethical practices when Creating Web/digital communications.

Unit Essential Questions	**Unit Enduring Understandings**
• What type of data can be contained in a list? • What does it mean to loop or iterate through an array? • How important is the use of data structures in 21st century life?	• An array is the most basic storage type which allows all more complex storage devices to function. • Finding an element in an array is the most fundamental processing purpose. • Outputting formats must be designed with the focus on the end users interface.

Career Readiness Practices

CRP2. Apply appropriate academic and technical skills.

CRP4. Communicate clearly and effectively and with reason.

CRP6. Demonstrate creativity and innovation.

CRP7. Employ valid and reliable research strategies.

CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.

CRP11. Use technology to enhance productivity.

Evidence of Learning

Summative Assessment: Each unit will involve the creation of numerous computer programs. A culminating project-based program will be included in each unit.

Formative Assessments

- Tests and quizzes
- Discussions
- Individual practice
- Explanation of examples
- Daily homework assignments
- Computer Programs

Lesson Plans

Lesson	Timeframe
Lesson 1 Introduction to Arrays	2 days
Lesson 2 Creating an Array	2 days
Lesson 3 Adding/Removing Elements from an Array	2 days
Lesson 4 Looping an Array	3 days

Lesson 5 Iterating an Array	2 days
Lesson 6 Introduction to Objects	3 days
Lesson 7 Introduction to Sets	4 days
Lesson 8 Introduction to Maps	4 days
Lesson 9 Introduction to Grids	4 days

Teacher Notes:

- 6 days allocated for formative/summative assessments
- Students can seek input from their peers and teachers throughout collaborative assignments and activities

Curriculum Development Resources

CodeHS :Introduction to Computer Science Course

Content Area Unit Name	English Language Arts, Mathematics, Science, Social Studies, World Language, Practical and Fine Arts, Business
Interdisciplinary Connections	Mathematics, Technology, and English Arts, Science
Core Instructional Materials including digital tools	Textbooks, Classroom Resources, Digital Tools
21st Century Themes and Skills	For information related to the 12 Career Ready Practices follow the links below: http://www.state.nj.us/education/cccs/2014/career/CareerReadyPractices.pdf Personal Financial Literacy 9.1 http://www.state.nj.us/education/cccs/2014/career/91.pdf Career Awareness, Exploration, and Preparation 9.2 http://www.state.nj.us/education/cccs/2014/career/92.pdf Career and Technical Education 9.3 http://www.state.nj.us/education/cccs/2014/career/93.pdf
8.1 Educational Technology **8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming**	K-2: Navigate provided URL'S, Use basic word processing to create and illustrate a simple story, Work collaboratively with peers on project, Use digital tools to explore an issue and design solution for a problem, Identify how technology improves life, Use digital tools to design an approach to solving problems. 3-5: Peers collaborate to produce text about current events; Understand the consequences for inappropriate use of technology and social media, Apply engineering designs to data collection and solutions, Understand how technology evolves based on need and cultural influences. 6-8: Select appropriate technology and applications to create publication on global topic, Use technology and social media responsibly, Employ a wide range of digital resources to collect data and form solutions, Identify the forces that come into play for further development of technology; apply engineering design process to real world problems. 9-12: Create and edit multi-page document for public presentation.

Considerations for classified students:

Classroom Instruction:

- All instruction for classified students will be guided by the students' Individualized Education Plan (IEP).
- Regular education teachers will be responsible for differentiating instruction for classified students based on the instructional modifications listed in the IEP.
- In the case of General Education - Supported Instruction (GE-SI) Classes, the special education teacher will be responsible for support in modifying the curriculum for the students, informing the class room teacher of the modifications, and directing instructional aide(s) to provide support accordingly.
- Grading will be done collaboratively by the regular and special education teachers.

Modifications:

- Modifications include but are not limited to:
  Extra time for assignments, modified classwork/homework assignments based on disability, preferential seating, study guides, copies of class notes, assistive technology and rewording/repeating or clarifying directions.

In-class Assessments:

- All assessments are to be in line with students' IEPs. In-class support teachers should modify tests for classified students. Tests may be given in the regular education classroom or completed with the inclusion teacher in another location with additional time. Students may be tested separately according to the IEP.
- Assessment grades may be modified based on a student's disability and in accordance with their IEP.

Considerations for English Language Learners (ELLs):

Classroom Instruction:

- Instruction for ESL students will be guided by their WIDA English Language Proficiency level. Teachers should receive this level from the ESL teacher assigned to the building.
- General education teachers will be responsible for differentiating instruction for ELLs with the assistance of the ESL teacher that promotes language, literacy and content learning.
- Sheltered Instruction Observation Protocol (SIOP)
  http://siop.pearson.com/about-siop/
  The following 8 components provide all teachers with lesson planning and instructional strategies that support language and learning goals for all students. This approach to teaching aligns with preparing students with college and career ready skills.
  The SIOP Model components:
  1. Lesson Preparation
  2. Building Background
  3. Comprehensible Input
  4. Strategies
  5. Interaction

6. Practice and Application
7. Lesson Delivery
8. Review and Assessment

- In the case of Content-Based ESL (CBE), the ESL teacher and the general education teacher will be responsible for identifying language objectives and additional instructional strategies that improve proficiency in English and academic success of ELLs. Instructional strategies and the necessary scaffolds to promote student learning will be shared with the general education teacher for daily lessons that are aligned to District Curricula, CCSS, and WIDA Standards. The general Education teacher and ESL teacher will be co-teachers for a pre-determined amount of classroom instruction.
- Grading will be done collaboratively by the regular and ESL teachers.

Modifications: The following are possible modifications but are not limited to this list –

- Direct instruction, small group or pullout, about the contrasting letter sound correspondences, syllabication patterns and morphology in English supported with connections to their native language, native language text and/or resources, graphic organizers, visuals, sentence starters/ sentence frames, cloze activities, modeling, working with a partner, timeline and phrase wall and adapted text (in English) or specific sections of the original text, highlighted/bold-faced words within text.
- Draw pictures instead of writing/speaking.
- Match drawings with new vocabulary that might correspond.
- Work in small group or pairs with their English Only (EOs) peers for authentic content language talk and grade level modeling.
- Write simple sentences instead of complex sentences that demonstrates an understanding of academic language particular to specific content.
- Match simple sentences with new vocabulary that might apply to edit sentences.
- Have students provide examples/explanations of main idea in simple sentences. Revisions show an attempt to improve Language Control by embedding academic content vocabulary and Linguistic Complexity by expanding and varying sentence structures and using correct punctuation.
- Draw pictures instead of writing/speaking about seasonal changes. Match drawings with new vocabulary (adjective word wall, content word walls) that might correspond.
- Provide multiple opportunities for authentic speech acts to practice language skills and develop English fluency.
- Total Physical Response (TPR) to model critical thinking skills like analyze and synthesize.
- Study Guides

In Class Assessments:

- All formative and summative assessments will include modifications that support student's English Proficiency level. ESL teachers will collaborate with regular education teachers to provide appropriate differentiation for assessing ELLs.

Considerations for At Risk Students:

- At Risk students are identified by the I&RS committee in each school. The committee works to understand the reasons behind the student's low performance level in school and to create and implement a plan that is carried out by a variety of staff members in the building.
- Teachers with At Risk students are notified by the I&RS committee and provided with a copy of the plan and a timeframe for assessing the growth of the student. There are academic as well as behavioral goals that are listed for the students with recommended strategies unique to each individual.
- Classroom teachers are to follow the plan using instructional strategies that will help the student improve his/her performance while applying appropriate behavioral strategies consistent with the needs of the student.
- Teachers will report student progress to the I&RS committee within the specified timeframe for the plan.

Classroom instruction:

- Teachers will use differentiated instruction for At Risk students as they do for all students in their class. The strategies would be guided by the I&RS plan and be consistent with the student's ability and learning modality.

Modifications:

- Clarify all assignments and place specific timeframes for completion. Provide student with opportunity for one on one time for clarification.
- Set clear expectations for all assignments, in and outside of class. Keep expectations within the framework of the I&RS plan.
- Use positive reinforcement for all successes. Hold student to defined consequences for not completing work.
- Provide time outside the normal class time for completion of work. Not completing assignments is unacceptable, all assignments will be completed.

In Class Assessments:

- At Risk students should receive any modifications listed in their I&RS plan.
- If necessary, students should be provided with extended time to complete assessments.

Considerations for Gifted Students:

- Teachers will use differentiated instruction for Gifted Students as they do for all students in their class.
- Assignments and assessments can be planned and implemented with input from the student.
- Gifted students will be provided with the opportunity to demonstrate their knowledge through a variety of platforms.
- Teachers will have the latitude to provide assignments with the individual student's ability in mind.